POSTECH-MLG-2011-009

Machine Learning Group

Department of Computer Science, POSTECH

Bayesian Matrix Co-Factorization: Variational Algorithm and Cramér-Rao Bound [a]

Jiho Yoo and Seungjin Choi

Machine Learning Lab
Department of Computer Science
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu
Pohang 790-784, Korea
Email: {zentasis,seungjin}@postech.ac.kr

Abstract

Matrix factorization is a popular method for collaborative prediction, where unknown ratings are predicted by user and item factor matrices which are determined to approximate a user-item matrix as their product. Bayesian matrix factorization is preferred over other methods for collaborative filtering, since Bayesian approach alleviates overfitting, integrating out all model parameters using variational inference or sampling methods. However, Bayesian matrix factorization still suffers from the cold-start problem where predictions of ratings for new items or of new users' preferences are required. In this paper we present *Bayesian matrix co-factorization* as an approach to exploiting side information such as content information and demographic user data, where multiple data matrices are jointly decomposed, i.e., each Bayesian decomposition is coupled by sharing some factor matrices. We derive variational inference algorithm for Bayesian matrix co-factorization. In addition, we compute Bayesian Cramér-Rao bound in the case of Gaussian likelihood, showing that Bayesian matrix co-factorization indeed improves the reconstruction over Bayesian factorization of single data matrix. Numerical experiments demonstrate the useful behavior of Bayesian matrix co-factorization in the case of cold-start problems.

[a] to be presented at ECML-PKDD-2011.

Contents

Machine Learning Lab
Department of Computer Science
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu
Pohang 790-784, Korea

1 Introduction

Matrix factorization is a method for seeking a low-rank latent structure of data, approximating the data matrix as a product of two or more factor matrices. Matrix factorization is popular for collaborative prediction, where unknown ratings are predicted by user and item factor matrices which are determined to approximate a user-item matrix as their product [6, 8, 4, 5, 11, 2]. Probabilistic matrix factorization was introduced in [8], in which a linear model with Gaussian observations was considered to learn user-specific and term-specific latent features, which became equivalent to the minimization of sum-of-squared errors with quadratic regularization terms. Bayesian approaches to matrix factorization are proposed based on the approximate inference such as the variational inference [4] or sampling [7], since the exact inference for the probabilistic model is intractable. Bayesian matrix factorization is preferred over other methods for collaborative filtering, since Bayesian approach alleviates overfitting by integrating out all model parameters.

Collaborative prediction algorithms suffer from the cold-start problem, where the users or items do not have sufficient number of given ratings. The cold-start problem commonly occurs in applying collaborative prediction in the practical problems because new users and new items, which has no previously given ratings, are continuously added to the system. Moreover, the users do not have high intention to rate the items remain in the system with small number of ratings of their own. The prediction accuracy of the collaborative prediction algorithm is seriously degraded because the algorithm only exploits the ratings given by the target users or items. To remedy the cold-start problem, efficient use of side information, such as item content information and user demographic information is crucial. Constrained probabilistic matrix factorization [8] is a representative method to incorporate side information into collaborative prediction based on matrix factorization, but it does not have clear relationship between the entity-relationship model of the whole data, so exploiting various kind of side information is not straight-forward.

Matrix co-factorization provides a way to systematically exploit the side information from the additional matrices. Matrix co-factorization jointly decomposes multiple data matrices, where each decomposition is coupled by sharing some factor matrices. Matrix co-factorization has been used to improve the performance of matrix factorization by incorporating knowledge in the additional matrices, such as label information [16], link information [17], and inter-subject variations [3]. One of the advantages of the matrix co-factorization is that it can be applied for the general entity-relationship models of the target data and the additional data [9, 14], where the factor matrices correspond to the entities and the input matrices correspond to the relationships of the model. Since the entity-relationship model is a fundamental tool to model the relational data, this simple mapping between the entity-relationship model and the co-factorization model enables the straight-forward use of various kind of side information, especially for the cold-start problems where both the user side information and the item side information are required. Recently, Cramér-Rao bound (CRB) was computed for matix co-factorization with Gaussian likelihood on compressed sensing, showing that CRB is improved over matrix factorization, in the sense of reconstruction error when side information is incorporated into co-factorization [15].

We present a Bayesian matrix co-factorization (BMCF) to exploit side information, such as content information and user demographic data, into collaborative prediction problem to remedy the cold-start problems. We derive variational inference algorithm for BMCF. Sampling method is another possible approach for the BMCF [10], however the posterior computation requires storing multiple number of samples which is not appropriate for the large-scale collaborative prediction problems. A variational Bayesian approach for matrix co-factorization was mentioned in [13] without any details, so in this paper we provide the descriptions of the specific probabilistic model and the computation of variational posteriors, hyperparameters, and the predictive distributions.

In addition, we compute Bayesian Cramér-Rao bound (BCRB) for the BMCF model. BCRB provides a lower bound on the variance of any parametric estimators, even for the unbiased ones [12]. We compute the bound for the reconstruction error based on the BCRB, to show that BMCF indeed improves the reconstruction over Bayesian matrix factorization (BMF) of single data matrix. Numerical experiments confirm the improvements of the theoretical performance from BCRB, and demonstrate the useful behavior of BMCF in cold-start cases.

2 Bayesian Matrix Co-Factorization

The simplest case of matrix co-factorization deals with two input matrices, namely, the user-item rating matrix $\boldsymbol{X} \in \mathbb{R}^{I \times J}$ and the user-demographic information matrix $\boldsymbol{Y} \in \mathbb{R}^{I \times K}$. The input matrices are decomposed into the products of the following form,

$$\begin{aligned} \boldsymbol{X} &\approx \boldsymbol{U}^\top \boldsymbol{V}, \\ \boldsymbol{Y} &\approx \boldsymbol{U}^\top \boldsymbol{W}, \end{aligned}$$

where $\boldsymbol{U} \in \mathbb{R}^{D \times I}$ is the user factor matrix, $\boldsymbol{V} \in \mathbb{R}^{D \times J}$ is the item factor matrix, and $\boldsymbol{W} \in \mathbb{R}^{D \times K}$ is the demographic factor matrix. The user factor matrix $\boldsymbol{U}$ is shared in both decompositions, which makes it to be learned from the side information $\boldsymbol{Y}$ as well as the target ratings $\boldsymbol{X}$. The use of information in $\boldsymbol{Y}$ makes possible to predict meaningful ratings where $\boldsymbol{X}$ has extremely small number of given ratings.

To set up the probabilistic model for the co-factorizations, each element of the input matrices is modeled with the additive Gaussian noises, such as

$$\begin{aligned} x_{ij} &= \boldsymbol{u}_i^\top \boldsymbol{v}_j + \varepsilon_{ij}^{(x)}, \text{ for all } (i,j) \in \mathcal{O}^{(x)}, \\ y_{ik} &= \boldsymbol{u}_i^\top \boldsymbol{w}_k + \varepsilon_{ik}^{(y)}, \text{ for all } (i,j) \in \mathcal{O}^{(y)}, \end{aligned}$$

where $\boldsymbol{u}_i$ represents the i-th column of $\boldsymbol{U}$, $\boldsymbol{v}_j$ represents the j-th column of $\boldsymbol{V}$, and $\boldsymbol{w}_k$ represents the k-th column of $\boldsymbol{W}$. $\mathcal{O}^{(x)}$ and $\mathcal{O}^{(y)}$ denote the set of all indices of observed elements in $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. The additive noise $\varepsilon_{ij}^{(x)}$ and $\varepsilon_{ik}^{(y)}$ are modeled with the Gaussian distribution, such as

$$\begin{aligned} \varepsilon_{ij}^{(x)} &\sim \mathcal{N}(\varepsilon_{ij}^{(x)} | 0, \rho^{(x)}), \\ \varepsilon_{ik}^{(y)} &\sim \mathcal{N}(\varepsilon_{ik}^{(y)} | 0, \rho^{(y)}), \end{aligned}$$

where $\mathcal{N}(x|\mu, \rho)$ represents the Gaussian distribution with mean μ and the variance ρ, and $\rho^{(x)}$ and $\rho^{(y)}$ represent the noise variances for $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. Then, the likelihood of the co-factorization is modeled as

$$\begin{aligned} p(\boldsymbol{X}, \boldsymbol{Y} | \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) &= p(\boldsymbol{X} | \boldsymbol{U}, \boldsymbol{V}) p(\boldsymbol{Y} | \boldsymbol{U}, \boldsymbol{W}) \\ &= \prod_{(i,j) \in \mathcal{O}^{(x)}} \mathcal{N}(x_{ij} | \boldsymbol{u}_i^\top \boldsymbol{v}_j, \rho^{(x)}) \prod_{(i,k) \in \mathcal{O}^{(y)}} \mathcal{N}(y_{ik} | \boldsymbol{u}_i^\top \boldsymbol{w}_k, \rho^{(y)}). \end{aligned}$$

The prior probabilities for the factor matrices are modeled with Gaussian,

$$\begin{aligned} p(\boldsymbol{U}) &= \prod_i \mathcal{N}(\boldsymbol{u}_i | \boldsymbol{0}, \boldsymbol{\Sigma}^{(u)}) = \prod_d \prod_i \mathcal{N}(u_{di} | 0, \rho_d^{(u)}), \\ p(\boldsymbol{V}) &= \prod_j \mathcal{N}(\boldsymbol{v}_j | \boldsymbol{0}, \boldsymbol{\Sigma}^{(v)}) = \prod_d \prod_j \mathcal{N}(v_{dj} | 0, \rho_d^{(v)}), \\ p(\boldsymbol{W}) &= \prod_k \mathcal{N}(\boldsymbol{w}_k | \boldsymbol{0}, \boldsymbol{\Sigma}^{(w)}) = \prod_d \prod_k \mathcal{N}(w_{dk} | 0, \rho_d^{(w)}), \end{aligned}$$

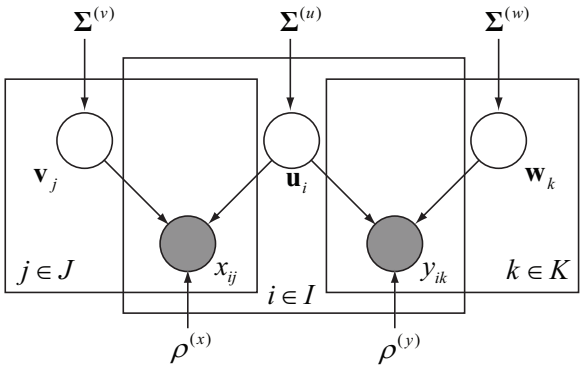

Figure 1: The graphical model representation of the Bayesian matrix co-factorizations, where a side information matrix $\boldsymbol{Y}$ is available with the target matrix $\boldsymbol{X}$.

where $\boldsymbol{\Sigma}^{(u)}$, $\boldsymbol{\Sigma}^{(v)}$ and $\boldsymbol{\Sigma}^{(w)}$ are the diagonal covariance matrices with the d-th diagonal element $\rho_d^{(u)}$, $\rho_d^{(v)}$, and $\rho_d^{(w)}$, respectively. Fig. 1 shows the graphical model representation of the probabilistic model.

We use the variational Bayesian approach to compute the posterior probability of each factor matrix. The lower-bound of the log of the marginal likelihood is computed by the Jensen's inequality with the functional $\mathcal{F}(q)$ of the auxiliary function $q(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})$, such as

$$\begin{aligned}
\log p(\boldsymbol{X}, \boldsymbol{Y}) &= \log \int\int\int q(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \frac{p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})}{q(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})} d\boldsymbol{U} d\boldsymbol{V} d\boldsymbol{W} \\
&\geq \int\int\int q(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \log \frac{p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})}{q(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})} d\boldsymbol{U} d\boldsymbol{V} d\boldsymbol{W} \\
&\equiv \mathcal{F}(q).
\end{aligned}$$

In the variational Bayesian framework, we assume that the auxiliary function is further factorized into

$$q(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) = q_u(\boldsymbol{U}) q_v(\boldsymbol{V}) q_w(\boldsymbol{W}),$$

leading to

$$\mathcal{F}(q_u, q_v, q_w) = \int\int\int q_u(\boldsymbol{U}) q_v(\boldsymbol{V}) q_w(\boldsymbol{W}) \log \frac{p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})}{q_u(\boldsymbol{U}) q_v(\boldsymbol{V}) q_w(\boldsymbol{W})} d\boldsymbol{U} d\boldsymbol{V} d\boldsymbol{W},$$

and $-\mathcal{F}(q_u, q_v, q_w)$ is referred to as *variational free energy.*

2.1 Updating Factor Matrices

In the variational Bayesian framework, the variational posteriors of the factor matrices $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{W}$ are computed with the following iterative updates,

$$q_u(\boldsymbol{U}) = \frac{1}{Z_u} \exp\left[\mathbb{E}_{V,W}\left\{\log p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})\right\}\right], \tag{1}$$

$$q_v(\boldsymbol{V}) = \frac{1}{Z_v} \exp\left[\mathbb{E}_{U,W}\left\{\log p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})\right\}\right], \tag{2}$$

$$q_w(\boldsymbol{W}) = \frac{1}{Z_w} \exp\left[\mathbb{E}_{U,V}\left\{\log p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})\right\}\right]. \tag{3}$$

To compute the variational posterior $q_u(\boldsymbol{U})$, the expectation over $\boldsymbol{V}$ and $\boldsymbol{W}$ is computed for the terms related to $\boldsymbol{U}$, which is written by

$$\begin{aligned}
& \mathbb{E}_{V,W}\left\{\log p(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})\right\} \\
= & -\frac{1}{2}\sum_i \left[\boldsymbol{u}_i^\top \left(\left(\boldsymbol{\Sigma}^{(u)}\right)^{-1} + \frac{1}{\rho^{(x)}} \sum_{\substack{j|(i,j)\\ \in\mathcal{O}^{(x)}}} \left\langle \boldsymbol{v}_j \boldsymbol{v}_j^\top \right\rangle + \frac{1}{\rho^{(y)}} \sum_{\substack{k|(i,k)\\ \in\mathcal{O}^{(y)}}} \left\langle \boldsymbol{w}_k \boldsymbol{w}_k^\top \right\rangle\right) \boldsymbol{u}_i\right] \\
& -\frac{1}{2}\sum_i \left[-2\left(\frac{1}{\rho^{(x)}} \sum_{\substack{j|(i,j)\\ \in\mathcal{O}^{(x)}}} x_{ij} \left\langle \boldsymbol{v}_j \right\rangle^\top + \frac{1}{\rho^{(y)}} \sum_{\substack{k|(i,k)\\ \in\mathcal{O}^{(y)}}} y_{ik} \left\langle \boldsymbol{w}_k \right\rangle^\top\right) \boldsymbol{u}_i\right] + C,
\end{aligned}$$

where $\langle\cdot\rangle$ represents the expectation. From (1), the derivation leads to the variational posterior of $\boldsymbol{U}$ in the following form,

$$q_u(\boldsymbol{U}) \sim \prod_i \mathcal{N}\left(\boldsymbol{u}_i | \overline{\boldsymbol{u}}_i^{(u)}, \boldsymbol{\Phi}_i^{(u)}\right),$$

where

$$\begin{aligned}
\overline{\boldsymbol{u}}_i^{(u)} &= \boldsymbol{\Phi}_i^{(u)} \left(\frac{1}{\rho^{(x)}} \sum_{j|(i,j)\in\mathcal{O}^{(x)}} x_{ij} \left\langle \boldsymbol{v}_j \right\rangle + \frac{1}{\rho^{(y)}} \sum_{k|(i,k)\in\mathcal{O}^{(y)}} y_{ik} \left\langle \boldsymbol{w}_k \right\rangle\right), \\
\left(\boldsymbol{\Phi}_i^{(u)}\right)^{-1} &= \left(\boldsymbol{\Sigma}^{(u)}\right)^{-1} + \frac{1}{\rho^{(x)}} \sum_{j|(i,j)\in\mathcal{O}^{(x)}} \left\langle \boldsymbol{v}_j \boldsymbol{v}_j^\top \right\rangle + \frac{1}{\rho^{(y)}} \sum_{k|(i,k)\in\mathcal{O}^{(y)}} \left\langle \boldsymbol{w}_k \boldsymbol{w}_k^\top \right\rangle.
\end{aligned}$$

As stated before, the user factor matrix is updated by using the side information matrix $\boldsymbol{Y}$, as well as the rating matrix $\boldsymbol{X}$, which enables the learning in the cold-start situation where $\boldsymbol{X}$ has no given ratings for some users.

The variational posteriors for the factor matrices $\boldsymbol{V}$ is computed from (2), which becomes

$$q_v(\boldsymbol{V}) = \prod_j \mathcal{N}(\boldsymbol{v}_j | \overline{\boldsymbol{u}}_j^{(v)}, \boldsymbol{\Phi}_j^{(v)}),$$

where

$$\begin{aligned}
\overline{\boldsymbol{u}}_j^{(v)} &= \boldsymbol{\Phi}_j^{(v)} \left(\frac{1}{\rho^{(x)}} \sum_{i|(i,j)\in\mathcal{O}^{(x)}} x_{ij} \left\langle \boldsymbol{u}_i \right\rangle\right), \\
\left(\boldsymbol{\Phi}_j^{(v)}\right)^{-1} &= \left(\boldsymbol{\Sigma}^{(v)}\right)^{-1} + \frac{1}{\rho^{(x)}} \sum_{i|(i,j)\in\mathcal{O}^{(x)}} \left\langle \boldsymbol{u}_i \boldsymbol{u}_i^\top \right\rangle.
\end{aligned}$$

Similarly from (3), the variational posterior of $\boldsymbol{W}$ is computed by

$$q_w(\boldsymbol{W}) = \prod_k \mathcal{N}(\boldsymbol{w}_k | \overline{\boldsymbol{u}}_k^{(w)}, \boldsymbol{\Phi}_k^{(w)}),$$

where

$$\begin{aligned}
\overline{\boldsymbol{u}}_k^{(w)} &= \boldsymbol{\Phi}_k^{(w)} \left(\frac{1}{\rho^{(y)}} \sum_{i|(i,k)\in\mathcal{O}^{(y)}} y_{ik} \left\langle \boldsymbol{u}_i \right\rangle\right), \\
\left(\boldsymbol{\Phi}_k^{(w)}\right)^{-1} &= \left(\boldsymbol{\Sigma}^{(w)}\right)^{-1} + \frac{1}{\rho^{(y)}} \sum_{i|(i,k)\in\mathcal{O}^{(y)}} \left\langle \boldsymbol{u}_i \boldsymbol{u}_i^\top \right\rangle.
\end{aligned}$$

The sufficient statistics for the above posteriors are easily computed by using the properties of the Gaussian distribution. The sufficient statistics for $\boldsymbol{u}_i$ are computed as

$$\begin{aligned} \langle \boldsymbol{u}_i \rangle &= \overline{\boldsymbol{u}}_i^{(u)}, \\ \left\langle \boldsymbol{u}_i \boldsymbol{u}_i^\top \right\rangle &= \boldsymbol{\Phi}_i^{(u)} + \overline{\boldsymbol{u}}_i^{(u)} \overline{\boldsymbol{u}}_i^{(u)\top}, \end{aligned}$$

and the sufficient statistics for $\boldsymbol{v}_j$ and $\boldsymbol{w}_k$ are computed in the similar forms.

2.2 Learning Hyperparameters

We use the empirical Bayes estimation to update hyperparameters $\rho^{(x)}$, $\rho^{(y)}$, $\boldsymbol{\Sigma}^{(u)}$, $\boldsymbol{\Sigma}^{(v)}$ and $\boldsymbol{\Sigma}^{(w)}$. The variational free energy $\mathcal{F}(q_u, q_v, q_w)$ is used to compute the point estimate of the hyperparameters.

Taking derivative of the variational free energy with respect to $\rho^{(x)}$ leads

$$\frac{\partial \mathcal{F}(q_u, q_v, q_w)}{\partial \rho^{(x)}} = -\frac{N^{(x)}}{2} \frac{1}{\rho^{(x)}} + \frac{1}{2(\rho^{(x)})^2} \sum_{(i,j) \in \mathcal{O}^{(x)}} \left\langle (x_{ij} - \boldsymbol{u}_i^\top \boldsymbol{v}_j)^2 \right\rangle,$$

where $N^{(x)}$ represents the total number of observed entries in the matrix $\boldsymbol{X}$. Then, $\rho^{(x)}$ is computed by

$$\rho^{(x)} = \frac{1}{N^{(x)}} \sum_{(i,j) \in \mathcal{O}^{(x)}} \left\{ x_{ij}^2 - 2 x_{ij} \langle \boldsymbol{u}_i \rangle^\top \langle \boldsymbol{v}_j \rangle + \mathrm{tr}\left(\left\langle \boldsymbol{u}_i \boldsymbol{u}_i^\top \right\rangle \left\langle \boldsymbol{v}_j \boldsymbol{v}_j^\top \right\rangle \right) \right\},$$

where $\mathrm{tr}(\cdot)$ represents the trace of the matrix. The update for $\rho^{(y)}$ is computed in the same way.

Taking derivative of $\mathcal{F}(q_u, q_v, q_w)$ with respect to $\rho_d^{(u)}$, which is the d-th diagonal element of $\boldsymbol{\Sigma}^{(u)}$, leads

$$\frac{\partial \mathcal{F}(q_u, q_v, q_w)}{\partial \rho_d^{(u)}} = -\frac{I}{2} \frac{1}{\rho_d^{(u)}} + \frac{1}{2 \left(\rho_d^{(u)} \right)^2} \left[\sum_i \left\langle \boldsymbol{u}_i \boldsymbol{u}_i^\top \right\rangle \right]_{dd},$$

and set this to be zero leads the update

$$\rho_d^{(u)} = \frac{1}{I} \left[\sum_i \left\langle \boldsymbol{u}_i \boldsymbol{u}_i^\top \right\rangle \right]_{dd}.$$

The above update is re-written for $\boldsymbol{\Sigma}^{(u)}$ in the following form,

$$\boldsymbol{\Sigma}^{(u)} = \frac{1}{I} \mathrm{ddiag} \left(\sum_i \left\langle \boldsymbol{u}_i \boldsymbol{u}_i^\top \right\rangle \right),$$

where $\mathrm{ddiag}(\boldsymbol{A})$ represents the diagonal matrix consisting of the diagonal elements of the matrix $\boldsymbol{A}$. The update for $\boldsymbol{\Sigma}^{(v)}$ and $\boldsymbol{\Sigma}^{(w)}$ are derived in the similar way.

2.3 Predictive Distribution

There are two kinds of prediction tasks in the collaborative prediction problem: the *hold-out* prediction and the *fold-in* prediction. In the hold-out prediction, we want to predict a

missing entry $x_{i^*j^*}$ in the input rating matrix $\boldsymbol{X}$, where $(i^*, j^*) \notin \mathcal{O}^{(x)}$. Then, the predictive distribution is calculated as

$$\begin{aligned} p(x_{i^*j^*}|\boldsymbol{X}) &= \int p(x_{i^*j^*}|\boldsymbol{U},\boldsymbol{V}) q_u^*(\boldsymbol{U}) q_v^*(\boldsymbol{V}) d\boldsymbol{U} d\boldsymbol{V} \\ &= \mathcal{N}(x_{i^*j^*}|\left\langle \boldsymbol{u}_{i^*} \right\rangle^\top \left\langle \boldsymbol{v}_{j^*} \right\rangle, \rho^{(x)}). \end{aligned}$$

Therefore, the prediction becomes the product of corresponding columns of factor matrices, which is $\hat{x}_{i^*j^*} = \left\langle \boldsymbol{u}_{i^*} \right\rangle^\top \left\langle \boldsymbol{v}_{j^*} \right\rangle$.

In the fold-in prediction, we want to predict the rating value of new users or items. If we want to predict the rating $x_{i^+j^*}$ for the new user i^+, the predictive distribution is computed by

$$\begin{aligned} p(x_{i^+j^*}|\boldsymbol{X},\boldsymbol{x}_{i^+},\boldsymbol{Y},\boldsymbol{y}_{i^+}) &= \int\int\int p(x_{i^+j^*}|\boldsymbol{u}_{i^+},\boldsymbol{v}_{j^*}) p(\boldsymbol{u}_{i^+}|\boldsymbol{V},\boldsymbol{x}_{i^+},\boldsymbol{W},\boldsymbol{y}_{i^+}) \\ &\quad p(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y}) d\boldsymbol{U} d\boldsymbol{V} d\boldsymbol{W} d\boldsymbol{u}_{i^+}. \end{aligned}$$

The predictive distribution depends on the posterior distribution of the new factor, which is computed by using the Bayes' rule,

$$\begin{aligned} & \log p(\boldsymbol{u}_{i^+}|\boldsymbol{V},\boldsymbol{x}_{i^+},\boldsymbol{W},\boldsymbol{y}_{i^+}) \\ = \; & \log p(\boldsymbol{x}_{i^+}|\boldsymbol{V},\boldsymbol{u}_{i^+}) + \log p(\boldsymbol{y}_{i^+}|\boldsymbol{W},\boldsymbol{u}_{i^+}) + \log p(\boldsymbol{u}_{i^+}) + C \\ = \; & \log \mathcal{N}(\boldsymbol{u}_{i^+}|\overline{\boldsymbol{u}}_{i^+}^{(u)}, \boldsymbol{\Phi}_{i^+}^{(u)}), \end{aligned}$$

where

$$\begin{aligned} \left(\boldsymbol{\Phi}_{i^+}^{(u)}\right)^{-1} &= \left(\boldsymbol{\Sigma}^{(u)}\right)^{-1} + \frac{1}{\rho^{(x)}} \sum_{j|(i^+,j)\in\mathcal{O}^{(x)}} \left\langle \boldsymbol{v}_j \boldsymbol{v}_j^\top \right\rangle + \frac{1}{\rho^{(y)}} \sum_{k|(i^+,k)\in\mathcal{O}^{(y)}} \left\langle \boldsymbol{w}_k \boldsymbol{w}_k^\top \right\rangle, \\ \overline{\boldsymbol{u}}_{i^+}^{(u)} &= \boldsymbol{\Phi}_{i^+}^{(u)} \left(\frac{1}{\rho^{(x)}} \sum_{j|(i^+,j)\in\mathcal{O}^{(x)}} x_{i^+j} \left\langle \boldsymbol{v}_j \right\rangle + \frac{1}{\rho^{(y)}} \sum_{k|(i^+,k)\in\mathcal{O}^{(y)}} y_{i^+k} \left\langle \boldsymbol{w}_k \right\rangle \right). \end{aligned}$$

This posterior indicates that the prediction is computed based on the observed ratings in $\boldsymbol{x}_{i^+}$ and the additional information $\boldsymbol{y}_{i^+}$, which makes the prediction in the cold-start situation possible. The unknown ranking in the fold-in case is predicted with the posterior distribution by $x_{i^+j^*} = \left\langle \boldsymbol{u}_{i^+} \right\rangle^\top \left\langle \boldsymbol{v}_{j^*} \right\rangle$, where $\left\langle \boldsymbol{u}_{i^+} \right\rangle = \overline{\boldsymbol{u}}_{i^+}^{(u)}$.

2.4 BMCF for General Cases

So far we considered the simplest example of the co-factorization, which has three entities: user, item, and user demographic information, and two relationships: user-item ratings and user-demographic information. We generalize the results for the arbitrary entity-relationship model by mapping the entities to the factor matrices and relationships to the input matrices. In this way, co-factorization model is directly induced from the entity-relationship model of data, which enables straight-forward use of various kinds of side-information.

The entity-relationship model consists of entities, attributes for the entities, and relationships between the entities. For the one-to-one correspondence between the entity-relationship model and the co-factorization model, we eliminate the use of attributes by modeling them as a separate entity having relationship with the corresponding entity. Then, we use the entity-relationship model consists of the set of entities $\mathcal{E}$ and the set of relationships $\mathcal{R}$. The co-factorization model is built with the input matrices $\boldsymbol{X}^{(a,b)}$ for all relationships $(a,b) \in \mathcal{R}$ and and the factor matrix $\boldsymbol{U}^{(a)}$ for all entities $a \in \mathcal{E}$. If we use the indices for a-th entity as i_a, the matrix co-factorization model is written by

$$x_{i_a i_b}^{(a,b)} = \boldsymbol{u}_{i_a}^{(a)\top} \boldsymbol{u}_{i_b}^{(b)} + \varepsilon_{i_a i_b}^{(a,b)} \text{ for all } (a,b) \in \mathcal{R}, (i_a, i_b) \in \mathcal{O}^{(a,b)},$$

where $\mathcal{O}^{(a,b)}$ represents the set of all observed entries in $\boldsymbol{X}^{(a,b)}$, and we used the additive Gaussian noise $\varepsilon^{(a,b)}_{i_a i_b}$ having distribution

$$\varepsilon^{(a,b)}_{i_a i_b} \sim \mathcal{N}(\varepsilon^{(a,b)}_{i_a i_b} | 0, \rho^{(a,b)}),$$

where $\rho^{(a,b)}$ represents the noise variance, which leads the Gaussian likelihood. The prior for each factor matrix $\boldsymbol{U}^{(a)}$ is modeled as Gaussian with zero mean and the variance $\rho^{(a)}_d$. The graphical model representation of the general matrix co-factorization is shown in the Fig. 2. The probabilistic model, update of factor matrices and hyperparameters, and the predictive distributions are summarized in Table 1.

3 Bayesian Cramér-Rao Bounds for Bayesian Matrix Co-Factorization

The Cramér-Rao Bound (CRB) places a lower bound on the variance of unbiased estimator for the deterministic parameters [1], as the inverse of the Fisher information matrix $\mathcal{F}$, which is written by,

$$\mathbb{E}\left\{(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top\right\} \geq \mathcal{F}^{-1},$$

where $\boldsymbol{\theta}$ is the estimated parameter and $\hat{\boldsymbol{\theta}}$ is the true value for it. Each element of the Fisher information matrix is computed by

$$\mathcal{F}_{ij} = \mathbb{E}_{\boldsymbol{x}}\left\{-\frac{\partial^2 \log p(\boldsymbol{x}|\boldsymbol{\theta})}{\partial\theta_i \partial\theta_j}\right\}.$$

The computation of Fisher information matrix mainly depends on the likelihood of the model.

On the other hand, the Bayesian Cramer-Rao bound (BCRB) or Posterior Cramer-Rao Bound [12] uses a different form of the Fisher information matrix, which depends on the joint probability of the observation and the parameters,

$$\mathcal{F}_{ij} = \mathbb{E}_{\boldsymbol{x},\theta}\left\{-\frac{\partial^2 \log p(\boldsymbol{x}, \boldsymbol{\theta})}{\partial\theta_i \partial\theta_j}\right\}. \tag{4}$$

In this case we use the prior probability, as well as the likelihood, to compute the Fisher information matrix, and the expectation is also taken over the parameters. The benefit of

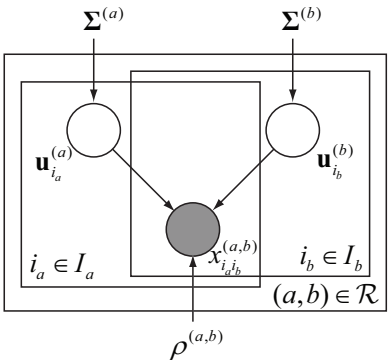

Figure 2: The graphical model representation of the Bayesian matrix co-factorizations in the general case.

Table 1: Model and the algorithms for the BMCF in general cases. We denote set of all input matrices as $\mathcal{X}$ and set of all factor matrices as $\mathcal{U}$. $I^{(a)}$ represents the number of columns in the factor matrix $\boldsymbol{U}^{(a)}$, and $N^{(a,b)}$ represents the number of observed entries in the input matrix $\boldsymbol{X}^{(a,b)}$.

Likelihood	$p(\mathcal{X}\|\mathcal{U}) = \prod_{(a,b)\in\mathcal{R}} \prod_{(i_a,i_b)\in\mathcal{O}^{(a,b)}} \mathcal{N}(x_{i_a i_b}^{(a,b)} \| \boldsymbol{u}_{i_a}^{(a)\top} \boldsymbol{u}_{i_b}^{(b)}, \rho^{(a,b)})$
Prior	$p(\boldsymbol{U}^{(a)}) = \prod_{i_a} \mathcal{N}(\boldsymbol{u}_{i_a}^{(a)} \| \boldsymbol{0}, \boldsymbol{\Sigma}^{(a)}) = \prod_d \prod_i \mathcal{N}(u_{di_a} \| 0, \rho_d^{(a)})$
Posterior	$q_a(\boldsymbol{U}^{(a)}) \sim \prod_{i_a} \mathcal{N}\left(\boldsymbol{u}_{i_a}^{(a)} \| \overline{\boldsymbol{u}}_{i_a}^{(a)}, \boldsymbol{\Phi}_{i_a}^{(a)}\right)$, where $\overline{\boldsymbol{u}}_{i_a}^{(a)} = \boldsymbol{\Phi}_{i_a}^{(a)} \left(\sum_{b\|(a,b)\in\mathcal{R}} \sum_{i_b\|(i_a,i_b)\in\mathcal{O}^{(a,b)}} \frac{1}{\rho^{(a,b)}} x_{i_a i_b}^{(a,b)} \left\langle \boldsymbol{u}_{i_b}^{(b)} \right\rangle \right)$ $\left(\boldsymbol{\Phi}_{i_a}^{(a)}\right)^{-1} = \left(\boldsymbol{\Sigma}^{(a)}\right)^{-1} + \sum_{b\|(a,b)\in\mathcal{R}} \sum_{i_b\|(i_a,i_b)\in\mathcal{O}^{(a,b)}} \frac{1}{\rho^{(a,b)}} \left\langle \boldsymbol{u}_{i_b}^{(b)} \boldsymbol{u}_{i_b}^{(b)\top} \right\rangle$
Sufficient statistics	$\left\langle \boldsymbol{u}_{i_a}^{(a)} \right\rangle = \overline{\boldsymbol{u}}_{i_a}^{(a)}$ $\left\langle \boldsymbol{u}_{i_a}^{(a)} \boldsymbol{u}_{i_a}^{(a)\top} \right\rangle = \boldsymbol{\Phi}_{i_a}^{(a)} + \overline{\boldsymbol{u}}_{i_a}^{(a)} \overline{\boldsymbol{u}}_{i_a}^{(a)\top}$
Parameters	$\rho^{(a,b)} = \frac{1}{N^{(a,b)}} \sum_{(i_a,i_b)\in\mathcal{O}^{(a,b)}} \left\{ \left(x_{i_a i_b}^{(a,b)}\right)^2 - 2 x_{i_a i_b}^{(a,b)} \left\langle \boldsymbol{u}_{i_a}^{(a)} \right\rangle^\top \left\langle \boldsymbol{u}_{i_b}^{(b)} \right\rangle \right\}$ $+ \frac{1}{N^{(a,b)}} \sum_{(i_a,i_b)\in\mathcal{O}^{(a,b)}} \left\{ \mathrm{tr}\left(\left\langle \boldsymbol{u}_{i_a}^{(a)} \boldsymbol{u}_{i_a}^{(a)\top} \right\rangle \left\langle \boldsymbol{u}_{i_b}^{(b)} \boldsymbol{u}_{i_b}^{(b)\top} \right\rangle \right) \right\}$ $\boldsymbol{\Sigma}^{(a)} = \frac{1}{I^{(a)}} \mathrm{ddiag}\left(\sum_{i_a} \left\langle \boldsymbol{u}_{i_a}^{(a)} \boldsymbol{u}_{i_a}^{(a)\top} \right\rangle \right)$
Prediction	$x_{i_a^* i_b^*} = \left\langle \boldsymbol{u}_{i_a^*}^{(a)} \right\rangle^\top \left\langle \boldsymbol{u}_{i_b^*}^{(b)} \right\rangle$ In the fold-in case, using $\left\langle \boldsymbol{u}_{i_a^*}^{(a)} \right\rangle = \boldsymbol{\Phi}_{i_a^*}^{(a)} \left(\sum_{c\|(a,c)\in\mathcal{R}} \left(\frac{1}{\rho^{(a,c)}} \sum_{i_c\|(i_a^*,i_c)\in\mathcal{O}^{(a,c)}} x_{i_a^* i_c} \left\langle \boldsymbol{u}_{i_c} \right\rangle \right) \right)$ $\left(\boldsymbol{\Phi}_{i_a^*}^{(a)}\right)^{-1} = \left(\boldsymbol{\Sigma}^{(a)}\right)^{-1} + \sum_{c\|(a,c)\in\mathcal{R}} \left(\frac{1}{\rho^{(a,c)}} \sum_{i_c\|(i_a^*,i_c)\in\mathcal{O}^{(a,c)}} \left\langle \boldsymbol{u}_{i_c}^{(c)} \boldsymbol{u}_{i_c}^{(c)\top} \right\rangle \right)$

using BCRB over CRB is that the BCRB is known to provide a lower bound on the variance of any parametric estimators, even for the unbiased ones [12]. In this section we use the BCRB to show the improvement of theoretical bounds of the proposed co-factorization model over the standard matrix factorization model.

3.1 Computation of Fisher Information Matrix

To compute the BCRB for the matrix co-factorization model, we rearrange the factor matrices to be a parameter vector. For example, if we have two factor matrices $\boldsymbol{U}^{(a)}$ and $\boldsymbol{U}^{(b)}$, the parameter vector $\boldsymbol{\theta}$ becomes

$$\boldsymbol{\theta} = [\boldsymbol{u}_1^{(a)\top} \cdots \boldsymbol{u}_{I^{(a)}}^{(a)\top} \boldsymbol{u}_1^{(b)\top} \cdots \boldsymbol{u}_{I^{(b)}}^{(b)\top}]^\top,$$

where $I^{(a)}$ represents the number of columns in the factor matrix $\boldsymbol{U}^{(a)}$. Then, each element of the Fisher information matrix is computed as (4). The log joint probability of BMCF is computed as the sum of log-likelihood and log priors (Table 1).

3.1.1 Case I $\left(\frac{\partial^2 \log p(\mathcal{X},\mathcal{U})}{\partial U^{(a)} \partial U^{(b)}}\right)$:

In this case, we take the first derivative of the log joint probability with respect to a parameter $u^{(a)}_{d^* i^*_a}$ in the factor matrix $\boldsymbol{U}^{(a)}$, which becomes,

$$\begin{aligned}
&\frac{\partial \log p(\mathcal{X},\mathcal{U})}{\partial u^{(a)}_{d^* i^*_a}} \\
&= \sum_{\substack{c|(a,c)\in\mathcal{R} \\ i_c|(i^*_a,i_c)\in\mathcal{O}^{(a,c)}}} \left[\frac{1}{\rho^{(a,c)}} x^{(a,c)}_{i^*_a i_c} u^{(c)}_{d^* i_c} - \frac{1}{\rho^{(a,c)}} u^{(c)}_{d^* i_c} \left(\sum_d u^{(a)}_{d i^*_a} u^{(c)}_{d i_c}\right)\right] - \frac{1}{\rho^{(a)}_{d^*}} u^{(a)}_{d^* i^*_a},
\end{aligned}$$

where $\mathcal{X}$ is the set of all input matrices and $\mathcal{U}$ is the set of all factor matrices. If we take the second derivative with respect to the parameter from the different factor matrix $\boldsymbol{U}^{(b)}$, which is $u^{(b)}_{d^+ i^+_b}$, it is written as

$$\frac{\partial^2 \log p(\mathcal{X},\mathcal{U})}{\partial u^{(a)}_{d^* i^*_a} \partial u^{(b)}_{d^+ i^+_b}} = \sum_{(i^*_a, i^+_b)\in\mathcal{O}^{(a,b)}} \left[-\frac{1}{\rho^{(a,b)}} u^{(a)}_{d^* i^*_a} u^{(b)}_{d^+ i^+_b}\right],$$

for $d^+ \neq d^*$. If $d^+ = d^*$, the second derivative is written as

$$\frac{\partial^2 \log p(\mathcal{X},\mathcal{U})}{\partial u^{(a)}_{d^* i^*_a} \partial u^{(b)}_{d^* i^+_b}} = \sum_{\substack{(i^*_a, i^+_b) \\ \in\mathcal{O}^{(a,b)}}} \left[\frac{1}{\rho^{(a,b)}} \left(x^{(a,b)}_{i^*_a i^+_b} - \sum_d u^{(a)}_{d i^*_a} u^{(b)}_{d i^+_b}\right) - \frac{1}{\rho^{(a,b)}} u^{(a)}_{d^* i^*_a} u^{(b)}_{d^* i^+_b}\right].$$

The expectations of above second derivatives vanish, so the elements of Fisher information matrix corresponding to the part also become zero.

3.1.2 Case II $\left(\frac{\partial^2 \log p(\mathcal{X},\mathcal{U})}{\partial U^{(a)} \partial U^{(a)}}\right)$:

The second derivative with respect to the element $u^{(a)}_{d^+ i^+_a}$ from the same factor matrix $\boldsymbol{U}^{(a)}$ vanishes if $i^+_a \neq i^*_a$. If $i^+_a = i^*_a$ and $d^+ \neq d^*$,

$$\frac{\partial^2 \log p(\mathcal{X},\mathcal{U})}{\partial u^{(a)}_{d^* i^*_a} \partial u^{(a)}_{d^+ i^*_a}} = -\sum_{c|(a,c)\in\mathcal{R}} \frac{1}{\rho^{(a,c)}} \sum_{i_c|(i^*_a,i_c)\in\mathcal{O}^{(a,c)}} u^{(c)}_{d^* i_c} u^{(c)}_{d^+ i_c},$$

but the expectation of it vanishes.

The only nonzero second-derivative value is arisen if we differentiate with the same element from the same matrix, that is, in the case of $i^+_a = i^*_a$ and $d^+ = d^*$, which becomes

$$\frac{\partial^2 \log p(\mathcal{X},\mathcal{U})}{\partial u^{(a)}_{d^* i^*_a} \partial u^{(a)}_{d^* i^*_a}} = -\sum_{c|(a,c)\in\mathcal{R}} \frac{1}{\rho^{(a,c)}} \sum_{i_c|(i^*_a,i_c)\in\mathcal{O}^{(a,c)}} \left(u^{(c)}_{d^* i_c}\right)^2 - \frac{1}{\rho^{(a)}_{d^*}}.$$

The Fisher information matrix is computed as

$$\mathbb{E}_{X,U}\left\{-\frac{\partial^2 \log p(\mathcal{X},\mathcal{U})}{\partial u^{(a)}_{d^* i^*_a} \partial u^{(a)}_{d^* i^*_a}}\right\} = \sum_{c|(a,c)\in\mathcal{R}} \frac{N^{(a,c)}_{i^*_a} \rho^{(c)}_{d^*}}{\rho^{(a,c)}} + \frac{1}{\rho^{(a)}_{d^*}}, \tag{5}$$

where $N^{(a,c)}_{i^*_a}$ represents the number of observed entries in the i^*_a-th column of the matrix $\boldsymbol{X}^{(a,c)}$. Because the only nonzero values come from the differentiating with the same parameter, the Fisher information matrix becomes a diagonal matrix.

If we use the standard matrix factorization, where there exist only two entities $\{a, c\}$ and one relationship, the diagonal elements of Fisher information matrix is computed as

$$\mathbb{E}_{X,U}\left\{-\frac{\partial^2 \log p(\mathcal{X},\mathcal{U})}{\partial u^{(a)}_{i^*_a d^*} \partial u^{(a)}_{i^*_a d^*}}\right\} = \frac{N^{(a,c)}_{i^*_a} \rho^{(c)}_{d^*}}{\rho^{(a,c)}} + \frac{1}{\rho^{(a)}_{d^*}},$$

which is obviously smaller than the Fisher information matrix of the matrix co-factorizations. Exploiting additional matrices in the co-factorization model increases the Fisher information matrix as the number of observed entries grows larger, which lowers the CRB (the inverse of the Fisher information matrix).

3.2 Computing Reconstruction Error

The major difficulty regarding BCRB in matrix factorization is the non-uniqueness of the matrix decomposition. Instead of directly using the BCRB, we consider the reconstruction error $\mathcal{E}_{ij}$, which is written as

$$\begin{aligned}\mathcal{E}_{ij} &= \mathbb{E}\left\{(x_{ij} - \hat{x}_{ij})^2\right\} \\ &= \mathbb{E}\left\{(\boldsymbol{u}_i^\top \boldsymbol{v}_j - \hat{\boldsymbol{u}}_i^\top \hat{\boldsymbol{v}}_j)^2\right\},\end{aligned}$$

where $\hat{x}_{ij}$, $\hat{\boldsymbol{u}}_i$, and $\hat{\boldsymbol{v}}_j$ are the ground-truth values, and x_{ij} is the predicted value from the estimated parameters $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$. Although the matrix decomposition is not uniquely determined, the reconstruction error is the same for the decompositions having the same likelihood. The reconstruction error is lower-bounded by using the BCRB, in a way that

$$\begin{aligned}\mathcal{E}_{ij} &= \mathbb{E}\left\{(\boldsymbol{u}_i^\top \boldsymbol{v}_j - \hat{\boldsymbol{u}}_i^\top \boldsymbol{v}_j + \hat{\boldsymbol{u}}_i^\top \boldsymbol{v}_j - \hat{\boldsymbol{u}}_i^\top \hat{\boldsymbol{v}}_j)^2\right\} \\ &= \mathbb{E}\left\{\boldsymbol{v}_j^\top (\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i^\top)(\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i^\top)^\top \boldsymbol{v}_j\right\} + \mathbb{E}\left\{\hat{\boldsymbol{u}}_i^\top (\boldsymbol{v}_j - \hat{\boldsymbol{v}}_j)(\boldsymbol{v}_j - \hat{\boldsymbol{v}}_j)^\top \hat{\boldsymbol{u}}_i\right\} \\ &\quad +2\mathbb{E}\left\{\boldsymbol{v}_j^\top (\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i)(\boldsymbol{v}_j - \hat{\boldsymbol{v}}_j)^\top \hat{\boldsymbol{u}}_i^\top\right\} \\ &\geq \mathbb{E}\left\{\boldsymbol{v}_j^\top \left[\mathcal{F}^{-1}\right]_{u_i} \boldsymbol{v}_j\right\} + \hat{\boldsymbol{u}}_i^\top \left[\mathcal{F}^{-1}\right]_{v_j} \hat{\boldsymbol{u}}_i + 2\mathbb{E}\left\{\boldsymbol{v}_j^\top (\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i)(\boldsymbol{v}_j - \hat{\boldsymbol{v}}_j)^\top \hat{\boldsymbol{u}}_i^\top\right\} \\ &= \hat{\boldsymbol{v}}_j^\top \left[\mathcal{F}^{-1}\right]_{u_i} \hat{\boldsymbol{v}}_j + \mathrm{tr}\left(\left[\mathcal{F}^{-1}\right]_{u_i} \left[\mathcal{F}^{-1}\right]_{v_j}\right) + \hat{\boldsymbol{u}}_i^\top \left[\mathcal{F}^{-1}\right]_{v_j} \hat{\boldsymbol{u}}_i,\end{aligned}$$

where $\left[\mathcal{F}^{-1}\right]_{u_i}$ represents the part of the inverse of the Fisher information matrix corresponding to the parameter $\boldsymbol{u}_i$, which is a diagonal matrix whose elements consists of the negative second derivatives of the joint probability with respect to $\boldsymbol{u}_i$.

4 Numerical Experiments

We performed two experiments with BMCF. First experiment computed the BCRB for the matrix co-factorization model and matrix factorization model, and compared them with the actual performance of the BMCF and BMF algorithms. Second experiment ran the BMCF and BMF algorithm for the collaborative prediction problem, where the number of given ratings were adjusted to simulate the cold-start situations.

4.1 BCRB Comparison on Synthetic Data

For the experiment comparing the reconstruction error computed from BCRB and the actual performance of the algorithm, we generated synthetic data with four entities $\mathcal{E} = \{1, 2, 3, 4\}$

and three relationships $\mathcal{R} = \{(1,2),(2,3),(3,4)\}$. The ground-truth factor matrices $\boldsymbol{U}^{(a)} \in \mathbb{R}^{5\times 100}$ were generated from the Gaussian distribution with variance 1. The relationship matrices were built from the factor matrices with additional Gaussian noise with variance 0.01. We chose the relation $(2,3)$ as the target matrix, where half of columns have 50% of observed entries, and the remaining columns have varying ratio of observed entries from 0% to 90%. The other relation matrices had 50% of observed entries. To show the benefit of the co-factorization, we compared the BCRB of the matrix co-factorization model with BCRB of matrix factorization model which used the target relationship matrix only. The actual performance was measured using the proposed BMCF algorithm and BMF algorithm. We used the Root Mean Squared Error (RMSE) for the performance measure, which is computed by

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|r_i - \overline{r}_i|^2},$$

where r_i represents the predicted value for the i-th test rating, $\overline{r}_i$ represents the true value, and N is the total number of test data points. Fig. 3 summarizes the result of the experiments. RMSE got better as the number of given ratings increases, both for the BCRB and the actual performance of the algorithm. BMCF had lower bound and performance compared to the BMF, and in this case the performance of BMCF was even lower than the theoretical lower-bound of BMF.

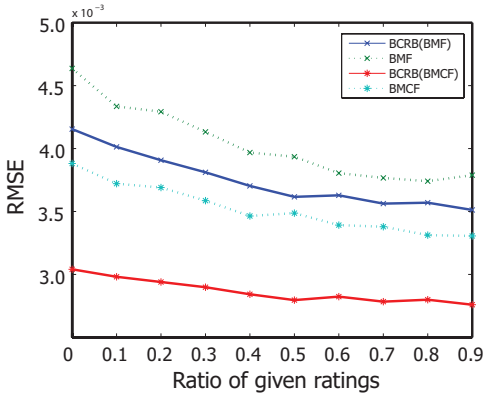

Figure 3: Comparison of the BCRB and the performance of the BMCF and BMF, averaged over 10 different trials.

4.2 Collaborative Prediction in the Cold-Start Situation

We applied the proposed BMCF for the collaborative prediction problem in the cold-start situations, and compared the performance with that of the BMF to show the benefit of the BMCF. We used MovieLens data, which consists of the ratings of 943 users for the 1682 movies for the test. The ratings are given by the integer score from 1 to 5. MovieLens data is packed with the additional user and movie information, which were used in the matrix co-factorization.

We constructed the additional information matrices of users and items in the following manner. User information consists of the age, gender, and occupation. The ages are partitioned into 5 groups, which are: under 20, 21 to 30, 31 to 40, 41 to 50, and over 51. The corresponding entry for the user was marked as the indicating value 1. The gender and occupations were coded in the similar way, indicating the user's gender and occupations by

Table 2: Average MAE and RMSE results for different number of given ratings for each test user. (a) Simulation of user cold-start case. (b) Simulation of user and item cold-start case. We eliminate all ratings for 100 randomly chosen items to simulate item cold-start case.

(a)	BMF		BMCF	
	MAE	RMSE	MAE	RMSE
0	2.5403	2.7767	0.8238	1.0140
5	0.8281	1.0618	0.7895	0.9941
10	0.8032	1.0205	0.7446	0.9424
15	0.7474	0.9558	0.7426	0.9314
20	0.7421	0.9496	0.7348	0.9254

(b)	BMF		BMCF	
	MAE	RMSE	MAE	RMSE
0	2.5098	2.7584	0.8843	1.0857
5	0.9333	1.2412	0.8332	1.0550
10	0.8956	1.1863	0.7778	0.9857
15	0.8991	1.1948	0.7716	0.9789
20	0.8618	1.1535	0.7527	0.9555

using the value 1. Movie information, which consists of the 18 category of the movie genres, was also marked in the similar way. In the experiments, we used the user information matrix and the item information matrix, as well as the user-movie rating matrix.

To simulate the cold-start situations for the users, we randomly chose 200 users in the dataset for the test users and generated the training data with different number of given ratings. Along with RMSE, we also computed the Mean Absolute Error (MAE) which is computed by

$$\text{MAE} \quad = \quad \frac{1}{N}\sum_{i=1}^{N}|r_i - \overline{r}_i|.$$

For each case, we randomly generated 10 different datasets, and ran the algorithm 10 times for each dataset with different initial values, so performance was measured 100 times for each case. Table 2(a) summarizes the averaged results for the experiments. BMF failed to predict the ratings when the test users have no ratings at all, however BMCF predicted fairly meaningful ratings for the case. The performance got better and better as the number of given ratings increases, but in all the cases, BMCF showed better performance than BMF, which showed the benefit of using side-information.

Another experiment was performed for the cases where some movies does not have any ratings at all. We randomly selected 100 movies from the dataset and eliminate all the ratings given for the movies. The averaged MAE and RMSE are summarized in Table 2(b). In this more severe condition, the performance of BMF was seriously degraded from the performance for the previous experiment. However, BMCF showed much better performance than BMF for all cases, slightly less than the results of the previous experiment. The use of the additional item information by using BMCF greatly helped the performance of the prediction, especially in this kind of item cold-start (as well as user cold-start) cases.

5 Conclusions

We have presented Bayesian matrix co-factorization (BMCF) as an approach to incorporating side information into collaborative prediction, where multiple data matrices are jointly decomposed, with some factor matrices shared over inter-related factorizations, in Bayesian setting. We have presented variational inference algorithm for updating factor matrices, in which variational posterior means and variances for factor matrices are iteratively updated. Hyperparameters are determined by maximizing the marginal likelihood. We have calculated Bayesian Cramér-Rao bound for the matrix co-factorization model, stressing that the co-factorization actually lowers the theoretical bound of the reconstruction error. Numerical experiments demonstrated that Bayesian matrix co-factorization yielded the lower BCRB

and improved the performance in collaborative prediction, compared to Bayesian matrix factorization. Especially in the case of cold start problems, Bayesian matrix co-factorization led to the satisfactory performance, while Bayesian matrix factorization failed to make proper predictions.

References

[1] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993.

[2] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[3] H. Lee and S. Choi. Group nonnegative matrix factorization for EEG classification. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Clearwater Beach, Florida, 2009.

[4] Y. J. Lim and Y. W. Teh. Variational Bayesian approach to movie rating prediction. In *Proceedings of KDD Cup and Workshop*, San Jose, CA, 2007.

[5] T. Raiko, A. Ilin, and J. Karhunen. Principal component analysis for large scale problems with lots of missing values. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 691–698, Warsaw, Poland, 2007.

[6] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*, Bonn, Germany, 2005.

[7] R. Salakhutdinov and A. Mnih. Bayesian probablistic matrix factorization using MCMC. In *Proceedings of the International Conference on Machine Learning (ICML)*, Helsinki, Finland, 2008.

[8] R. Salakhutdinov and A. Mnih. Probablistic matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20. MIT Press, 2008.

[9] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, Las Vegas, Nevada, 2008.

[10] A. P. Singh and G. J. Gordon. A Bayesian matrix factorization model for relational data. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, Catalina Island, CA, 2010.

[11] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, 10:623–656, 2009.

[12] P. Tichavsky, C. H. Muravchik, and A. Nehorai. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, 46(5):1386–1395, 1998.

[13] S. Williamson and Z. Ghahramani. Probabilistic models for data combination in recommender systems. In *NIPS-2008 Workshop on Learning from Multiple Sources*, Whistler, Canada, 2010.

[14] J. Yoo and S. Choi. Weighted nonnegative matrix co-tri-factorization for collaborative prediction. In *Proceedings of the 1st Asian Conference on Machine Learning (ACML)*, Nanjing, China, 2009.

[15] J. Yoo and S. Choi. Matrix co-factorization on compressed sensing. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Spain, 2011.

[16] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, Salvador, Brazil, 2005.

[17] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, Amsterdam, The Netherlands, 2007.

Acknowledgments

This work was supported by National Research Foundation (NRF) of Korea (2010-0014306, 2010-0018828, 2010-0018829), NIPA Program of Software Engineering Technologies Development and Experts Education, and NRF WCU Program (R31-10100).


POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교